

# ポリシー制御に基づく Web サービスの対話型トランザクションアーキテクチャの提案と評価

大谷 洋子<sup>†</sup> 青山 幹雄<sup>‡</sup>

南山大学 大学院 数理情報研究科<sup>†</sup>

南山大学 数理情報学部 情報通信学科<sup>‡</sup>

## 1. はじめに

Web サービストランザクションの仕様である WS-TX(WS-Transaction)[2]は, ACID 特性を緩和し, 複数のサービスを一つのビジネスタスクにまとめるスコープ機能, 処理を元の状態へ戻す補償機能により Web 上の処理に対応する. これを基に, ビジネスプロセス連携のためのトランザクション処理の要求を満たす対話型トランザクションアーキテクチャ(CTA)を提案し, 効果を示す.

## 2. Web サービストランザクションへの要求

### 2.1. 部分的な処理の実現

#### (1) 原子性の緩和

必ず成功しなくてはならない機能集合と, 選択的に必要な処理だけ成功すれば良いサービス集合を階層的に分離し管理する. 部分的に原子性を緩和し, 一貫性を保ちながら部分的な処理を行う.

#### (2) 独立性の緩和

長期リソースロックを回避するため, 独立性を緩和し, 個別の処理を並列に行い取捨選択することは, 契約の重複と補償によるビジネスリスクをもたらす. 個別の処理と全体の処理を分離して管理し, 不要な補償処理を行わないようにする.

### 2.2. 状況に応じた処理の実現

ビジネスでは必要な結果が期限内に得られなければ目的を達成できない場合がある. 処理完了の期限やサービスの優先度を設定し, ユーザの満足度の高い結果を得る処理を実現する.

## 3. トランザクションモデルへのアプローチ

上記の要求を満たすトランザクション処理を実現するために, 次の二つの方法を提案する.

### 3.1. スコープの定義

WS-TX のスコープ機能を用い, 管理するサービスの範囲を階層的に定める. 必ず成功すべき機能の集合を「親スコープ」と定義する. 同一機能を提供する選択可能なサービスの集合を「子スコープ」と定義する. 各子スコープ内で一つずつサービスと処理が確定すると, 親スコープの原子性を満たす. 子スコープが互いの処理内容を関知せずに, 処理の一貫性を保つ部分的な処理を実現する.

### 3.2. 対話モデルを用いたメッセージ交換

WS-TX のトランザクション処理は調整プロトコルの状態遷移に沿ったメッセージ交換により一貫性を保つ. 各状態で期限の通知や交渉を行い, 状況に対応した処理を実現するため, 対話モデルを用いる. 対話モデル[3]では対話を行う双方が対話制御部を持ち, 同じ対話ポリシー(CP)に沿ってメッセージを交換する. トランザクション管理と対話を実現する CP を作成し, 対話制御部によるトランザクション制御を実現する.

## 4. 対話型トランザクションアーキテクチャ

図 1 のように, CTA は WS-TX の Coordinator を基礎とし, 起動, 登録サービスがポート情報等を交換することでサービスと接続する. 対話制御部マネージャはスコープごとに対話制御部を生成する. 対話制御部を各スコープのサービスと関連付け, 対話制御とトランザクション管理を行う.

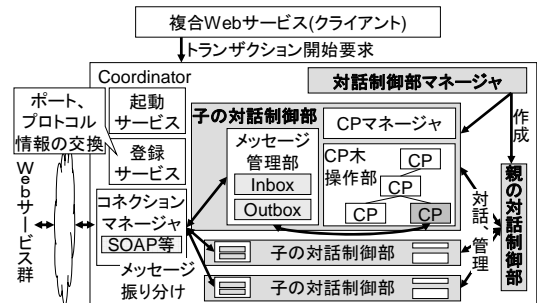


図 1 対話型トランザクションアーキテクチャ

### 4.1. トランザクションの適応型制御

ユーザから必要な機能, 期限, 優先度を取得し, ユーザの要求とビジネスの状況に適応するトランザクション処理を実現する. 優先度を用い, スコープごとに順位の高いサービスから処理することで, 満足度の高い結果を効率的に得る. 順位はアプリケーションが決めるものとする.

また, 個別の処理結果を全体の処理結果と分離し, 全体の処理に対する期限付きの予約とすることで不要な補償処理を回避する. 親と子の間で対話を行い, 親の対話制御部が管理する全体の期限と個別の処理期限を調整する.

## 5. 対話ポリシーの設計

図 2 のように, CP は要求される機能に応じて親と子で別々に定義する. スコープ内の管理に特化することで, サービスごとの対話の柔軟性とトランザクション処理全体の一貫性が両立できる.

Policy-based Conversation Architecture for Web Services Transaction Processing and its Evaluation

<sup>†</sup>Yoko Otani Graduate School of Mathematical Sciences and Information Engineering, Nanzan University

<sup>‡</sup>Mikio Aoyama Graduate School of Mathematical Sciences and Information Engineering, Nanzan University

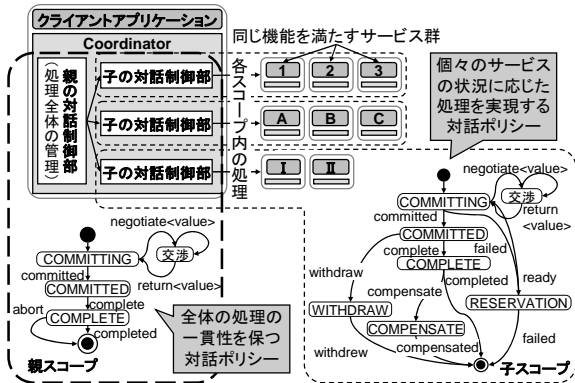


図 2 対話型トランザクション処理の制御

### 5.1. 親の対話制御部の対話ポリシー

親の対話制御部の対話ポリシーは図 3に示すように、トランザクション全体を管理する CP から、サービス選択の手順を定義した CP を呼び出す。呼び出す CP を変更することで処理内容に合うサービス選択手順を選択できる。子スコープがサービス選択 CP に沿って中間結果を確定した時点で、全体の管理 CP に戻り最終結果の確定を実行する。

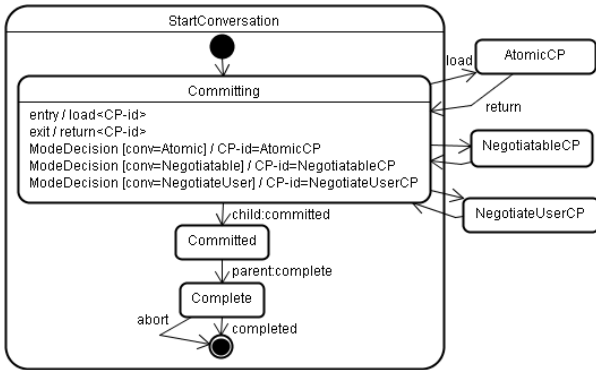


図 3 親の対話制御部の対話ポリシー例

### 5.2. 子の対話制御部の対話ポリシー

子の対話制御部の対話ポリシーはトランザクション処理中に複数のサービスと親との対話を同時に行う。そのため、図 4のように識別子によってメ

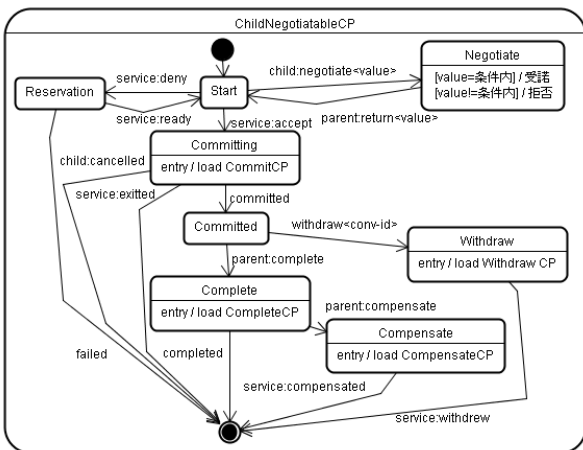


図 4 子の対話制御部の対話ポリシー例

ッセージを判別し、相手や状況に応じた対話用ポリシーを呼び出す。

## 6. 対話型トランザクションの評価

WS-TX と CTA を比較し、提案するアーキテクチャの有用性を評価する。旅行代理店の予約システムを例に、表 1の条件で処理を行った結果をビジネス状況の変化への対応の点から比較する。

表 1 旅行代理店の予約システム状況

スコープ	航空会社			ホテル			レンタカー	
サービス名	航空1	航空2	航空3	ホテルA	ホテルB	ホテルC	レンタカーI	レンタカーII
優先度	10	8	7	7	6	4	5	5
中間結果保持期限	7日	5日	5日	1日	3日	3日	7日	7日
サービスの初期状態	満席	○	○	○	○	○	○	○

上の条件に沿った場合、WS-TX の処理結果は、航空会社 2、ホテル A、レンタカー I となる。それに対し CTA では状況に応じて複数の処理結果が得られる。例として航空会社 1 に空席が生じた場合の処理の流れを図 5に示す。状況の変化に応じ、優先度によってホテル A より航空会社 1 を優先して確保する。このように CTA はビジネスプロセス連携の状況に対応する処理を実現できる。

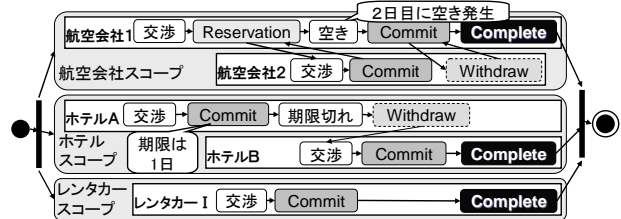


図 5 対話型トランザクションの予約フロー

## 7. 関連研究

文献[4]では、トランザクションの調整ルールを定義し、終了状態の比較により複合サービスのトランザクション要求の一貫性を保つ手法を提案しているが、複合処理手法は言及していない。

## 8. まとめと今後の課題

Web サービスに対応したトランザクション仕様である WS-TX と対話モデルを組み合わせた対話型トランザクションを提案した。今後は処理方法を定義するための対話ポリシーのルールを検討する。

### 参考文献

- [1] 大谷 洋子, 青山 幹雄, ポリシー制御に基づく Web サービスの対話型トランザクションアーキテクチャの提案と評価, SES2006, Oct. 2006, pp. 145-148.
- [2] F. Cabrera, et al., WS-Transaction, 2004, <http://www-128.ibm.com/developerworks/library/specification/ws-tx/>.
- [3] J. E. Hanson, et al., Conversation Support, 2003, <http://www.research.ibm.com/convsupport/>.
- [4] F. Montagut, et al., Augmenting Web Services Composition with Transactional Requirements, ICWS '06. Sept. 2006, pp. 91-98.